УТВЕРЖДЕН

ТАСП.62.01.11.000.003 34 01-ЛУ

ПРОГРАММНЫЙ КОМПЛЕКС

«КЛИЕНТСКАЯ ОПЕРАЦИОННАЯ СИСТЕМА»

ПК «СинтезМ-Клиент»

Руководство оператора

ТАСП.62.01.11.000.003 34 01

Листов 41

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

2017

Литера___

АННОТАЦИЯ

Данный документ является руководством оператора программного комплекса «Клиентская операционная система» ТАСП.62.01.11.000.003 34 01 (ПК «СинтезМ-Клиент»). Документ описывает назначение, условия выполнения и выполнение программы.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Назначение программы

Изделие представляет собой клиентскую операционную систему и предназначено для создания доверенной среды функционирования АРМ (рабочая станция, ноутбук) и применения в качестве гостевой операционной системы для виртуальных машин.

1.2. Функции программы

ПК «СинтезМ-Клиент» предназначен для выполнения следующих функций:

- обеспечения функционирования программного обеспечения, в качестве гостевой операционной системы, в виртуальной среде под управлением ПК «СинтезМ-Сервер» ТАСП.62.01.11.000.002 из состава ЗОС «СинтезМ»;
- обеспечения функционирования программной среды на рабочих станциях и ноутбуках;
- поддержки виртуальных машин (пользователей, серверов приложений, системы управления базами данных, системы хранения данных и подсистемы защиты информации);
- реализации двух вариантов установки на виртуальные машины: с графическим рабочим столом и без графического рабочего стола;
- создания доверенной среды с изоляцией процессов;
- поддержки настраиваемых посредством ПС «СинтезМ-АБ» механизмов защиты информации (идентификации и аутентификации; разграничения доступа; регистрации и учета; контроля целостности; антивирусной защиты; самотестирование компонентов ПСЗИ; защиты от НСД);
- поддержки синхронизации с сервером единого времени;

– поддержки файловых систем (NTFS, Ext2, Ext3, Ext4, Xfs, Vfat (fat32));

– функционирования средств для автоматизации деятельности пользователей (WEB-браузер, офисные средства (КП «СинтезМ-О» ТАСП.62.01.11.000.013), клиент СХД, проводник, средства обработки мультимедийных данных);

– поддержки USB-устройств, PCI-устройств и оптических дисков;

– поддержки работы с пакетом OpenSSL, функционирующим с применением алгоритмов криптографического преобразования ГОСТ 28147-89, ГОСТ 34.10-2012.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1. Требования к программным средствам

Функционирование изделия в качестве гостевой операционной системы виртуальных машин, функционирующих на физических серверах, в среде виртуализации под управлением ПК «СинтезМ-Сервер».

Для функционирования изделия на базе рабочей станции или ноутбука не требуется наличие предустановленного программного обеспечения.

При функционировании в составе АСЗИ изделие взаимодействует с:

1) комплексом программных средств защиты информации, функционирующих в среде операционной системы «СинтезМ» (КП «ПСЗИ «СинтезМ») ТАСП.62.01.11.000.004, в состав которого входят:

– программный комплекс «Сервер безопасности» ТАСП.62.01.11.000.005 (ПК «СинтезМ-СБ»);

– программное средство «Агент безопасности» ТАСП.62.01.11.000.006 (ПС «СинтезМ-АБ»);

– программный комплекс «Сервер управления доступом» ТАСП.62.01.11.000.007 (ПК «СинтезМ-ИПА»);

– программное средство «Система управления конфигурацией» ТАСП.62.01.11.000.008 (ПС «СинтезМ-УК»);

– программное средство «Репозиторий программного обеспечения» ТАСП.62.01.11.000.009 (ПС «СинтезМ-Репо»).

2) комплексом программных средств «Функциональные сервисы» (КПС «Функциональные сервисы») ТАСП.62.01.11.000.010, в состоящего из следующих компонентов:

– программное средство «Система управления базами данных» ТАСП.62.01.11.000.011 (ПС «СинтезМ-СУБД»);

– программное средство «Сервер приложений» ТАСП.62.01.11.000.012 (ПС «СинтезМ-СП»);

– комплекс программ «Офисные средства» ТАСП.62.01.11.000.013 (КП «СинтезМ-О»);

– комплекс программ «Контроль и управление функционированием» ТАСП.62.01.11.000.014 (КП «СинтезМ-КУФ»), состоящий из трех компонентов:

- программного комплекса «Сервер контроля и управления функционированием» ТАСП.62.01.11.000.015 (ПК «СинтезМ-КУФ.С»),
- программного комплекса «Прокси-сервер контроля и управления функционированием» ТАСП.62.01.11.000.016 (ПК «СинтезМ-КУФ.ПС»),
- программного средства «Агент контроля и управления функционированием» ТАСП.62.01.11.000.017 (ПС «СинтезМ-КУФ.А»);

– программное средство «Система хранения данных» ТАСП.62.01.11.000.018 (ПС «СинтезМ-СХД-01»);

– программное средство «Система хранения данных» ТАСП.62.01.11.000.019 (ПС «СинтезМ-СХД-02»);

– программное средство «Система резервного копирования» ТАСП.62.01.11.000.020 (ПС «СинтезМ-СРК»);

– комплекс программ «Защищенная электронная почта» ТАСП.62.01.11.000.021 (КП «СинтезМ-ЗЭП»), состоящий из:

- – программного средства «Сервер защищенной электронной почты» ТАСП.62.01.11.000.022 (ПС «СинтезМ-ЗЭП.С»);
- – программное средство «Клиент защищенной электронной почты» ТАСП.62.01.11.000.023 (ПС «СинтезМ-ЗЭП.К»).

3) аппаратно-программным средством «Персональный идентификатор пользователя» (АПС «СинтезМ-ТК/ПИ») ТАСП.62.09.20.190.001.

2.2. Требования к техническим средствам

Изделие предназначено для функционирования на рабочих станциях и ноутбуках, а также в качестве гостевой операционной системы виртуальных машин, функционирующих на физических серверах под управлением ПК «СинтезМ-Сервер» (60412878.00051-02) (вер.6.4).

Минимальные требования к ПК «СинтезМ-Сервер» (ТАСП.62.01.11.000.002) и принципы их расчета в зависимости от сложности АИС приведены в документации на ПК «СинтезМ-Сервер».

Для функционирования ПК «СинтезМ-Клиент» на рабочей станции или ноутбуке, необходимы технические средства со следующими минимальными характеристиками:

– процессор архитектуры x86_x64 с тактовой частотой не менее 2 ГГц;

– оперативная память не менее 4 ГБ;

– объем жесткого диска не менее 15 ГБ;

– сетевой контроллер с минимальной пропускной способностью 100 Мбит;

– монитор с разрешением 1024×768;

– клавиатура (рус./лат.);

– манипулятор «мышь».

Для функционирования ПК «СинтезМ-Клиент» же в качестве гостевой операционной системы виртуальных машин, необходимы технические средства со следующими минимальными характеристиками:

– процессор архитектуры x86_x64 с тактовой частотой не менее 2 ГГц;

– оперативная память не менее 4 ГБ;

– объем жесткого диска не менее 15 ГБ.

2.3. Функциональные ограничения на применение

Для функционирования изделия должен быть настроен сервер виртуализации под управлением ТАСП.62.01.11.000.002 ПК «СинтезМ-Сервер». Дополнительных функциональных ограничений не предъявляется.

2.3.1. Ограничения для совместно функционирующего ПО

1. Запрещается реализация в совместнофункционирующем ПО функций DHCP-сервера и PXE-сервера.

2. Запрещается реализация в совместнофункционирующем ПО функционала по модификации данных, хранящихся в ИПА, за исключением создания служебных учетных записей пользователей (не соответствующих пользователям Системы) и их настройке, осуществляемых в процессе развертывания ПО.

3. Запрещается реализация функционала по созданию локальных учетных записей ОС, за исключением создания учетных записей технологических пользователей, осуществляемого на этапе развертывания ПО.

4. Для ПО, функционирующего под учетной записью привилегированного пользователя, запрещается изменять следующие файловые объекты:

– все файлы в директории /var/lib/aide/sintez;

– файл /etc/init.d/aide_startup;

– файл counthash;

– файл /etc/aide.startup.conf;

– файл /etc/cron.d/counthash_timetable;

– файл /etc/aide.conf;

– файл /usr/bin/file_tree;

– файл /etc/sintez/integrity_client.conf;

– файл /etc/passwd;

– файл /etc/default/useradd;

– файл /etc/group;

– файл /etc/shadow;

– файл /etc/skel;

– файл /etc/selinux/targeted/seusers;

– файлы в директории /etc/selinux/targeted/contexts;

– файл /var/www/Sintez-Repo/web/.htaccess;

– файл /etc/httpd/conf.d/z-ovirt-engine-proxy.conf;

– файл /etc/aide.user-group.conf.

5. Привилегированному пользователю запрещается изменение параметра BIOS, содержащего описание источника загрузки образа ядра.

6. Привилегированному пользователю запрещается изменение образа ядра в существующем источнике.

7. Запрещается осуществлять запуск скрипта counthash.

8. Запрещается осуществлять изменение настроек запуска скрипта /etc/init.d/aide_startup посредством удаления и/или модификации ссылок на этот файл в подпапках папки /etc/rc.d/.

9. Запрещается реализация функционала по изменению таблицы системных вызовов ядра ОС.

10. Запрещается реализация функционала по передаче логина и пароля для получения TGT-билета Kerberos или подключения к Серверу управления виртуальными машинами.

11. Запрещается реализация функционала по редактированию файла /root/.ssh/authorized_keys на хостах системы.

12. Запрещается реализация функционала по редактированию содержимого следующих директорий:

а./etc/sysconfig/rsyslog;

б./etc/rsyslog.conf;

в.файлы из папки /etc/rsyslog.d/;

г.файлы из папки /etc/audit/.

13. Запрещается реализация остановки служб auditd и Rsyslog.

14. Запрещается предоставлять пользователю возможность выбора сервиса, который будет остановлен.

15. В случае реализации ПО модулей уровня ядра запрещается:

а.создавать блочное или символьное устройство, позволяющее осуществлять запись в себя для непривилегированных пользователей;

б.создавать блочное или символьное устройство, позволяющее осуществлять отображение содержимого файла устройства в контекст памяти пользовательского процесса для непривилегированных пользователей;

в.создавать RAM-based интерфейс ядра в виде файла или директории с правами, позволяющими осуществлять запись в себя для непривилегированных пользователей;

г.открывать UDP или NetLink сокет в режиме ядра;

д.отправлять сигналы в user-space процессы;

е.вызывать произвольный код из user-space.

16. ПО, устанавливаемому в ОС в виде драйверов, функционирующих в режиме ядра:

– драйвер не должен использовать механизм сетевых сокетов, а именно:

а.драйвер не должен использовать функционал Netlink сокетов;

б.драйвер не должен использовать функционал UDP сокетов.

– драйвер не должен добавлять новый системный вызов в таблицу системного вызова, скрипты сборки драйвера не должны модифицировать исходный код ОС для добавления системных вызовов

– драйвер не должен использовать функцию upcall

17. Программное обеспечение, функционирующее в рамках ЗОС «СинтезМ», должно пройти тематические исследования в части:

а.анализа уязвимостей, присутствующих в нем или в используемых им компонентах сторонней разработки и описанных в общедоступных источниках;

б.анализа уязвимостей и опасных возможностей, проводимого с использованием автоматизированного средства поиска уязвимостей, осуществляющего анализ исходных текстов ПО;

в.В ЗОС «СинтезМ» не должно присутствовать уязвимостей и опасных возможностей, которые потенциально могут быть использованы непривилегированным пользователем для влияния на механизмы защиты ЗОС «СинтезМ».

18. В АСЗИ, в рамках которой устанавливается ЗОС «СинтезМ», не допускается наличие хостов с ОС, отличной от ЗОС «Синтез» (версия 6.4) и ЗОС «СинтезМ».

Единственным средством обращения прикладного ПО к SELinux является библиотека libselinux. Таблица с описаниями функций (названия и назначения) для библиотеки libselinux приведена в приложении А.

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Начало работы

3.1.1. Включение рабочей станции

Включение рабочей станции производится путем нажатия на кнопку «Power», расположенную на передней панели.

3.1.2. Загрузка системы

Загрузка системы начинается сразу после включения рабочей станции, при этом по протоколу PXE на терминальную станцию по ЛВС производится загрузка операционной системы ПК «СинтезМ-Клиент» (рисунок 1).

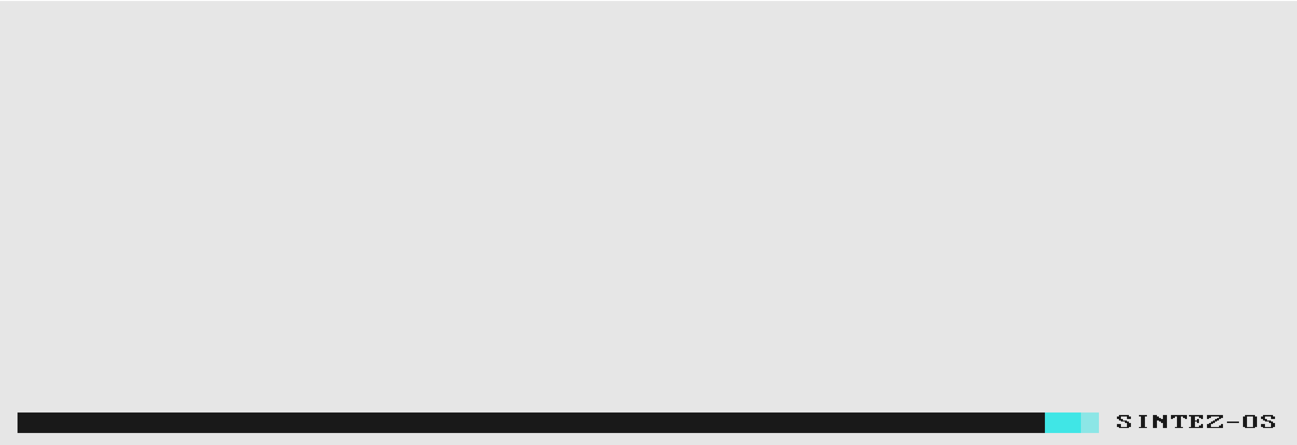


Рисунок 1 – Процесс загрузки системы

3.1.3. Авторизация пользователя на ВМ

После экрана загрузки ПК «СинтезМ-Клиент» появляется экран аутентификации пользователя ВМ (рисунок 2).

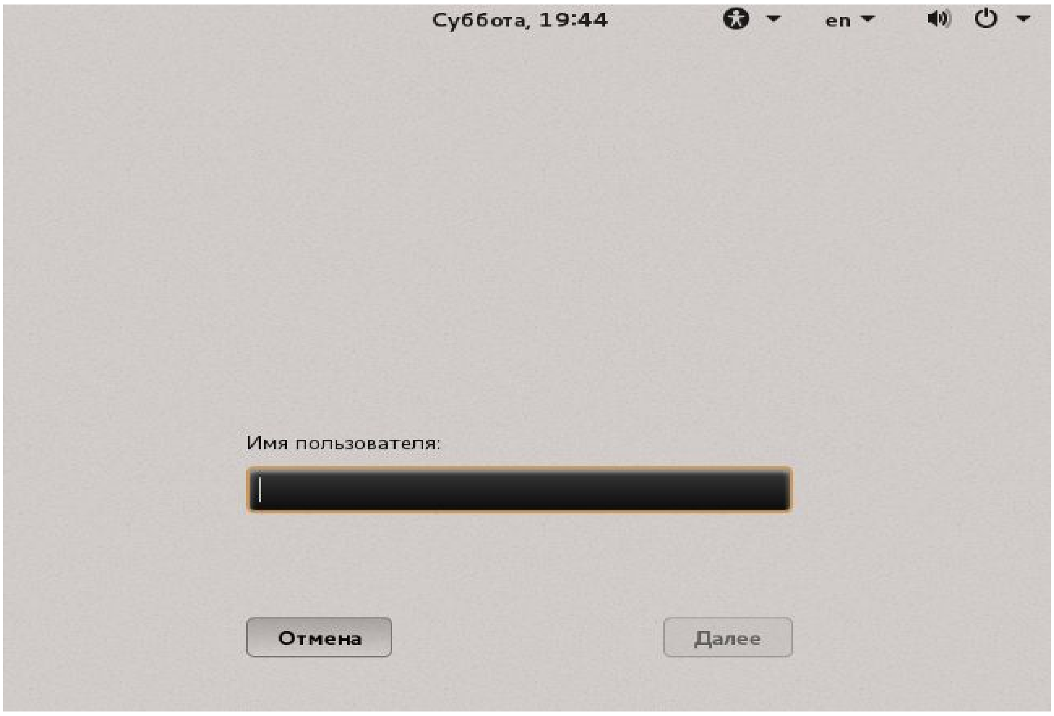


Рисунок 2 – Окно аутентификации пользователя

Для авторизации в системе необходимо вставить аутентифицирующий персональный идентификатор пользователя (АПС «СинтезМ-ТК/ПИ») (рисунок 3) в USB-порт рабочей станции.

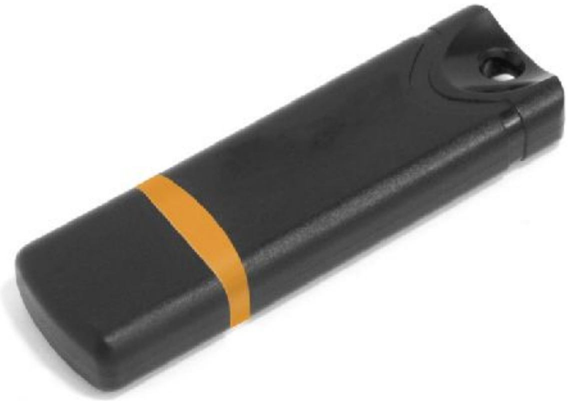


Рисунок 3. АПС «СинтезМ-ТК/ПИ»

Далее в поле «Пароль от карты» нужно ввести пароль пользователя системы, после этого нажать кнопку «Войти в систему».

При успешной авторизации пользователя на экране монитора должен отобразиться рабочий стол ВМ пользователя, как показано рисунке 4.

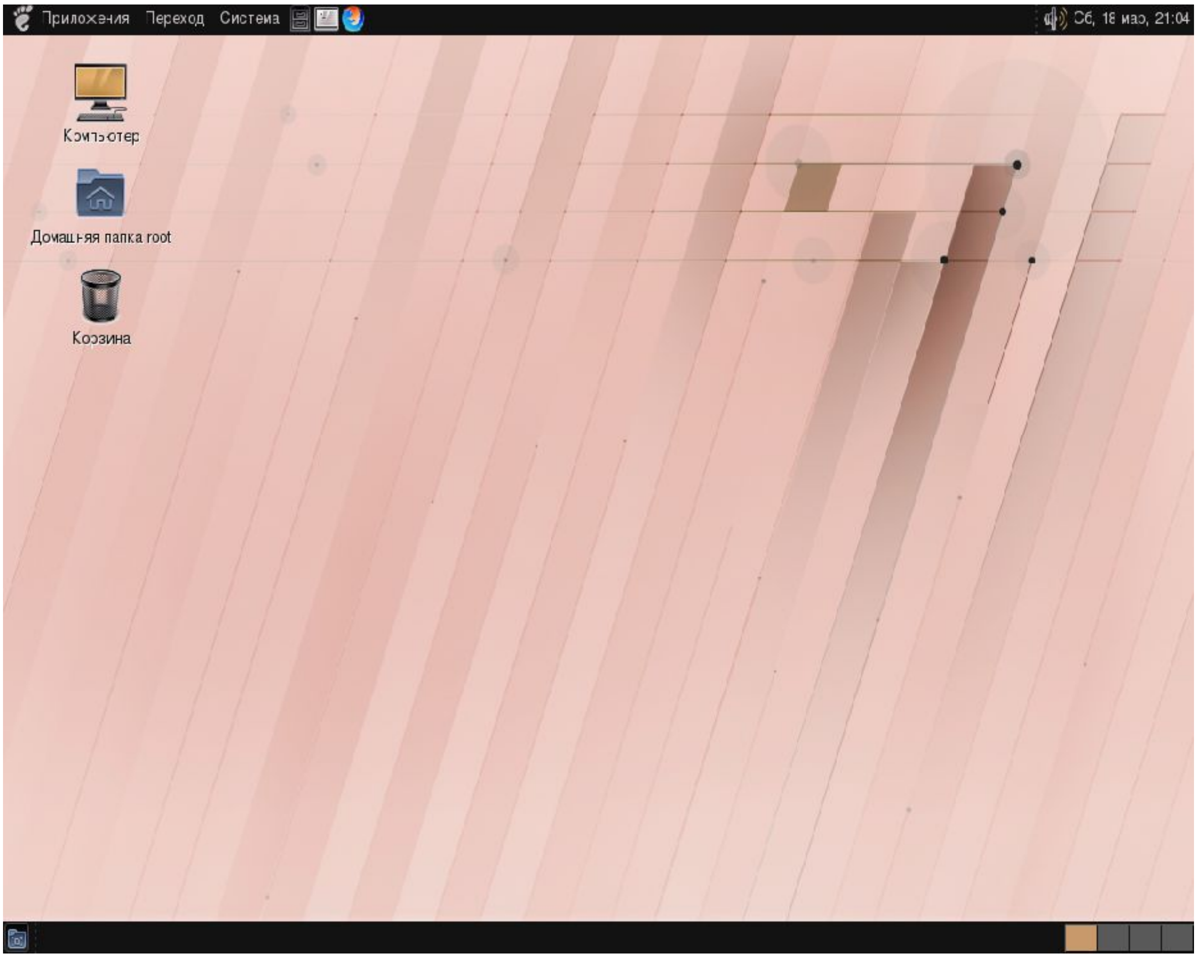


Рисунок 4 – Рабочий стол

3.2. Работа с встроенными приложениями

3.2.1. Браузер

Для запуска файлового менеджера необходимо запустить ярлык приложения в меню «Приложения» или значка быстрого доступа расположенном в верхней части рабочего стола.

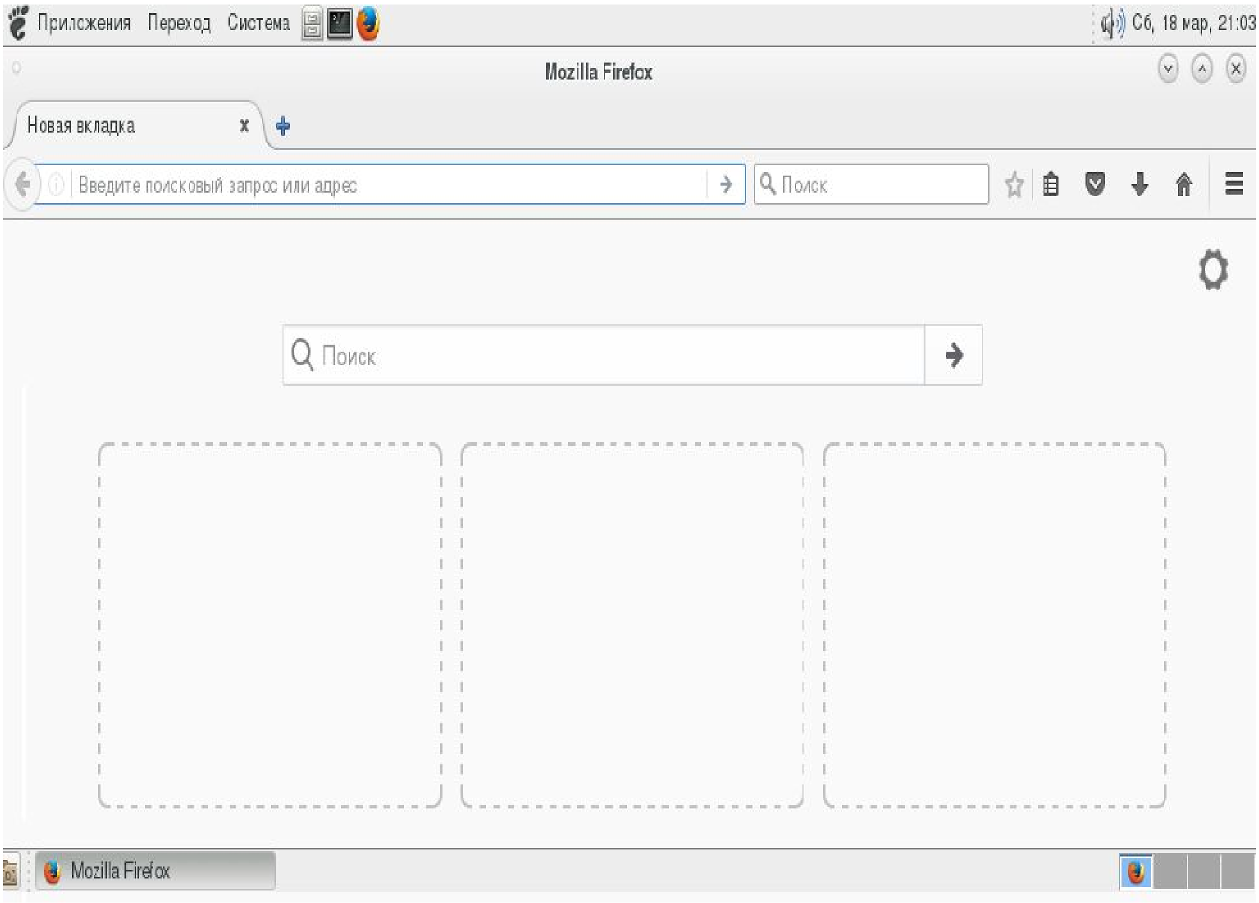
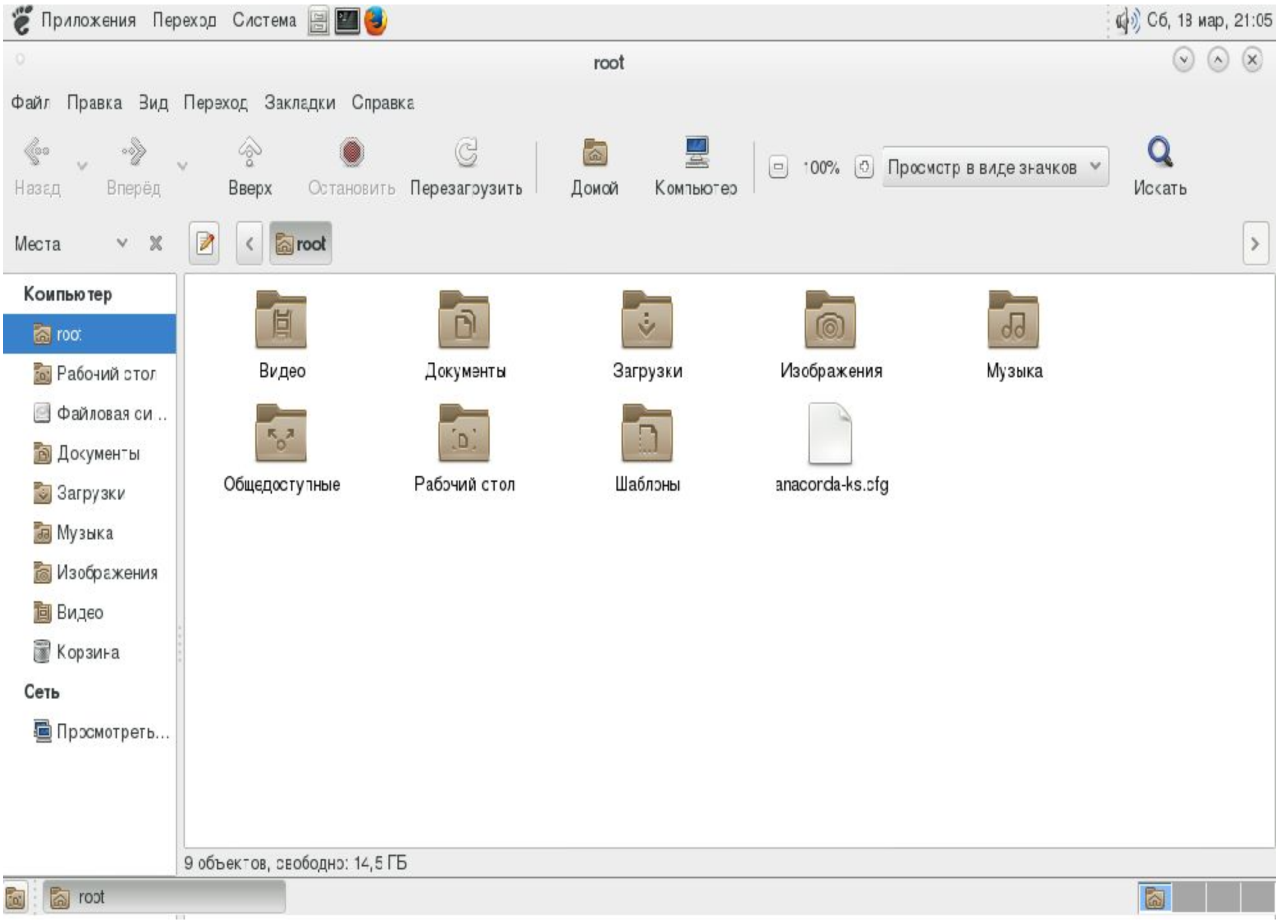


Рисунок 5– Окно браузера

Браузер – это программа, которая обеспечивает просмотр Интернет-сайтов на рабочей станции. Суть браузера заключается в том, чтобы обработать запрос пользователя, и загрузить запрашиваемую им веб-страницу. Главная функция браузера – это открытие веб-страниц сайтов. Сами же веб-страницы состоят из кода, который собственно и получает браузер с сервера, на котором находится сайт. Данный код браузер обрабатывает, и вы видите у себя на мониторе «готовую картинку».

3.2.2. Текстовый редактор

Для запуска файлового менеджера необходимо запустить ярлык приложения в меню «Приложения» расположенном в верхней части рабочего стола.



9 объектов, свободно: 14,5 ГБ

Рисунок 6 – Окно текстового редактора

Рабочее окно текстового редактора позволяет осуществлять форматирование текстовых и графических данных:

– настройка размера шрифта и цветов;

– настройка подсветки синтаксиса;

– функциональность копирования и вставки;

– отмена и повтор действий;

– откат файлов;

– удаленное редактирование файлов;

– поиск и замена;

– буфер обмена.

3.2.3. Файловый менеджер

Для запуска файлового менеджера необходимо запустить ярлык приложения в меню «Приложения» расположенном в верхней части рабочего стола.

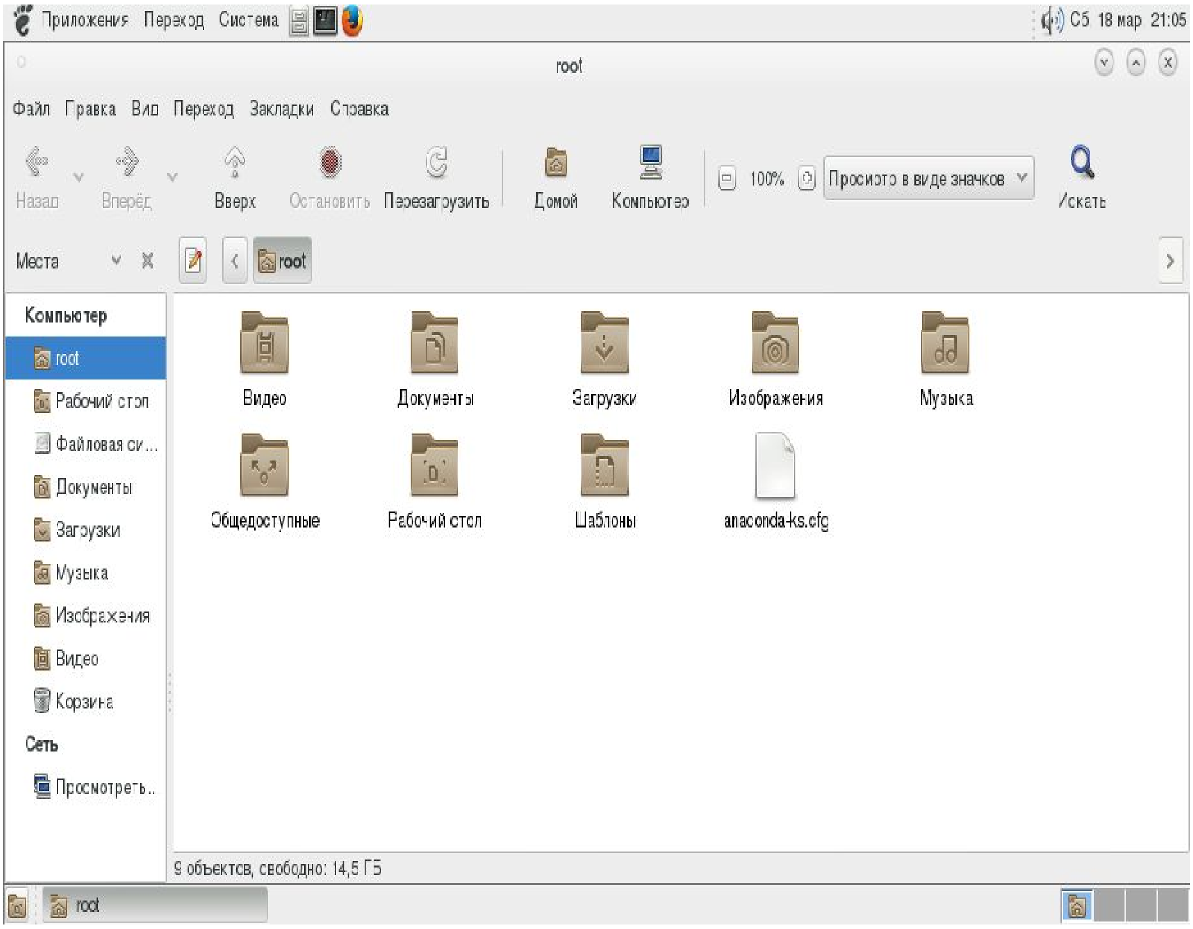


Рисунок 7 – Окно файлового менеджера

Рабочее окно файлового менеджера состоит из боковой панели и главного окна просмотра файлов и папок. В боковой панели можно отображать дерево каталогов, информацию о текущем элементе главной панели, список наиболее важных мест для посещения. Кроме того, файловый менеджер позволяет составлять и хранить в боковой панели заметки.

3.3. Выполнение функциональных задач

Функциональные задачи должностных лиц выполняются в соответствии с инструкцией по работе со специальным программным обеспечением.

3.4. Завершение работы

Для завершения работы с ВМ пользователя необходимо извлечь персональный идентификатор, подключенный к USB-порту рабочей станции.

3.4.1. Выключение рабочей станции

Перед выключением рабочей станции обязательно извлечь АПС «СинтезМ-ТК/ПИ».

Выключение рабочей станции производится аналогично процедуре включения путем зажатия на кнопки «Power», расположенной на передней панели рабочей станции.

Примечание: при выключении терминальной станции выполнение целевых задач на ВМ продолжается в фоновом режиме, ВМ пользователя не выключается.

3.4.2. Гашение экрана и блокировка системы

Реализация механизмов гашения экрана и блокировки системы осуществляется путем извлечения АПС «СинтезМ-ТК/ПИ» из USB-порта терминального клиента.

Для возобновления работы пользователя в ВМ необходимо произвести повторную авторизацию пользователя (п. 3.1.3).

4. СООБЩЕНИЯ ОПЕРАТОРУ

Сообщения могут быть информационные, предупреждающие или сообщения об ошибках. Возможные сообщения и необходимые действия оператора при их появлении описаны в разделе 3 настоящего документа.

ПРИЛОЖЕНИЕ А. ОПИСАНИЕ ФУНКЦИЙ ДЛЯ БИБЛИОТЕКИ LIBSELINUX

Название функции	Описание назначения
avc_add_callback	Функция регистрирует обработчик событий безопасности
avc_audit	Аудит предоставления или отказа разрешений в соответствии с политикой. Эта функция обычно вызывается avc_has_perm (3) после проверки разрешений, но также может быть вызвана непосредственно вызывающими методами, которые используют avc_has_perm_noaudit (3), для отделения проверки разрешений от аудита. Например, это разделение полезно, когда проверка разрешений должна выполняться под блокировкой, чтобы позволить блокировке высвободиться прежде вызова аудита кода.
avc_av_stats	Логирует статистику таблицы журналов. Регистрирует сообщение с информацией о размере и распределении таблицы векторов доступа. Для печати сообщения используется обратный вызов аудита.
avc_cache_stats	Получает кеш статистики доступа. Заполняет прилагаемую структуру информацией о деятельности AVC с момента последнего вызова avc_init (3) или avc_reset (3).
avc_cleanup	Удаляет неиспользуемые записи в кэше AVC. Ищет в таблице SID структуры SID с нулевым количеством ссылок и удаляет их вместе со всеми элементами AVC, которые ссылаются на них. Это можно использовать для высвобождения памяти в системе.
avc_compute_create	Вычисляет SID для маркировки нового объекта. Вызывает сервер безопасности, чтобы получить контекст для маркировки нового объекта.
avc_compute_member	Вычисляет SID для полиинстантации. Вызывает сервер безопасности, чтобы получить контекст для маркировки экземпляра объекта.
avc_context_to_sid	Получает SID для контекста. Просмотрите контекст безопасности ctx в таблице SID, создав новую запись, если ctx не найден. Хранит указатель на структуру SID в памяти, на которую ссылается sid, возвращая 0 при успехе или -1 при ошибке с установкой флага errno.
avc_context_to_sid_raw	Получает SID для контекста.
avc_destroy	Освобождает все структуры AVC. Уничтожает все структуры AVC и освобождает всю выделенную память. Удерживаемые пользователем блокировки,

Название функции	Описание назначения
	память и обратные вызовы аудита будут сохранены, но обратные вызовы событий безопасности не будут. Все SID будут аннулированы. Пользователь должен вызвать avc_init (3), если требуется дальнейшее использование AVC.
avc_get_initial_sid	Получает SID для исходного идентификатора безопасности ядра. Получает контекст для исходного идентификатора безопасности ядра, указанного по имени, с помощью security_get_initial_context (3), а затем вызывает avc_context_to_sid (3), чтобы получить соответствующий идентификатор безопасности.
avc_has_perm	Проверяет разрешения и выполняет любой соответствующий аудит. Осуществляет запрос AVC, чтобы определить, предоставлены ли запрашиваемые разрешения для SID (ssid, tsid), интерпретирует разрешения на основе tclass и вызове сервера безопасности в пропуске кеша, чтобы получить новое решение и добавить его в кеш. Обновляет aeref, чтобы ссылаться на запись AVC с принятыми решениями. Аудит предоставления или отказа в разрешениях в соответствии с политикой. Возвращает 0, если все запрошенные разрешения предоставлены, -1 с errno, установленным в EACCES, если какие-либо разрешения отклоняются или к другому значению при других ошибках.
avc_has_perm_noaudit	Проверяет разрешения, но не выполняет аудит. Проверяет AVC, чтобы определить, предоставлены ли запрашиваемые разрешения для пары SID (ssid, tsid), интерпретации разрешений на основе tclass и вызове сервера безопасности в пропуске кеша, для получения нового решения и добавления его в кеш. Обновляет aeref, чтобы ссылаться на запись AVC с принятыми решениями и возвращает копию решений в avd. Возвращает 0, если все запрашиваемые разрешения предоставляются, -1 с errno, установленным в EACCES, если разрешены какие-либо разрешения или другое значение при возникновении других ошибок. Эта функция обычно вызывается avc_has_perm (3), но также может быть вызвана непосредственно (самостоятельно), чтобы отделить проверку разрешений от аудита, например, в случаях, когда блокировка должна удерживаться для проверки, но должна быть осуществлена для аудита.
avc_init	Использует avc_open. Инициализирует AVC. Инициализация кеша вектора доступа. Возврат 0 при успехе или -1 с флагом errno, установленным при ошибке.

Название функции	Описание назначения
avc_netlink_acquire_fd	Создает сокет netlink и подключается к ядру.
avc_netlink_check_nb	Ожидает сообщения netlink из ядра.
avc_netlink_close	Закрывает сокет netlink.
avc_netlink_loop	Устанавливает netlink сокет fd. Позволяет приложению управлять сообщениями из сокета netlink в своем собственном основном цикле.
avc_netlink_open	Освобождает netlink сокет fd. Возвращает право собственности на сокет netlink к библиотеке.
avc_netlink_release_fd	Проверяет netlink сокет для новых сообщений.
avc_open	Инициализирует AVC. Эта функция идентична avc_init (3), за исключением того, что префикс сообщения установлен на «avc», и любые требуемые обратные вызовы должны быть указаны через selinux_set_callback (3).
avc_reset	Очищает кеш и сбрасывает статистику. Удаляет все записи из кэша и сбрасывает все статистические данные доступа (возвращаемые avc_cache_stats (3)) на ноль. Не влияет на отображение SID . Возвращает 0 при успехе, -1 с флагом errno, установленным при ошибке.
avc_sid_stats	Статистика таблицы SID журнала. Записывает сообщение с информацией о размере и распределении таблицы SID. Для вывода сообщения используется обратный вызов аудита.
avc_sid_to_context	Получает копию контекста, соответствующего SID. Возвращает копию контекста безопасности, соответствующего входному sid в памяти, на которое ссылается ctx.
avc_sid_to_context_raw	Получает копию контекста, соответствующего SID. Возвращает копию контекста безопасности, соответствующего входному sid в памяти, на которое ссылается ctx.
checkPasswdAccess	Используйте selinux_check_passwd_access (3) или предпочтительно selinux_check_access (3). Проверьте разрешение в классе passwd. Возвращает 0, если предоставлено или -1 в противном случае. Данная функция является "заглушкой".
context_free	Освобождает хранилище, используемое контекстом.
context_new	Возвращает новый контекст, инициализированный строкой контекста.
context_range_get	Получает указатель на диапазон.
context_range_set	Задает компонент диапазона. Возвращает ненулевое значение, если оно не выполнено.
context_role_get	Получает указатель на роль.
context_role_set	Установает компонент роли. Возвращает ненулевое значение, если оно не выполнено.
context_str	Возвращает указатель на строковое значение context_t. Действителен до следующего вызова

Название функции	Описание назначения
	context_str или context_free для одного и того же context_t*.
context_type_get	Получает указатель на тип.
context_type_set	Устанавливает тип компонента. Возвращает ненулевое значение, если оно не выполнено.
context_user_get	Получает указатель на пользователя.
context_user_set	Устанавливает пользовательский компонент. Возвращает ненулевое значение, если оно не выполнено.
fgetfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освобождаться через freecon.
fgetfilecon_raw	Обертка для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освобождаться через freecon.
fini_selinuxmnt	Очищает переменную selinuxmnt и освобождает выделенную память.
freecon	Освобождает память, выделенную для контекста любым вызовом get *.
freeconary	Освобождает память, выделенную для массива контекста, с помощью security_compute_user (3).
fsetfilecon	Обертка для API xattr - Устанавливает контекст файла.
fsetfilecon_raw	Обертка для API xattr - Устанавливает контекст файла.
getcon	Получает текущий контекст и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освободить через freecon (3).
getcon_raw	Получает текущий контекст и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освободить через freecon (3).
get_default_context	Получает контекст безопасности по умолчанию для сеанса пользователя для «пользователя», порожденного «fromcon», и устанавливает для него *newcon, чтобы ссылаться на него. Контекст будет одним из полномочий, разрешенных политикой, но выбор значения по умолчанию зависит от пользовательских настроек. Если ’fromcon’ равно NULL, по умолчанию используется текущий контекст. Возвращает 0 при успехе или -1 в противном случае. Вызывающий метод должен освобождаться через freecon.
get_default_context_with_level	То же, что и get_default_context (3), но используется предоставленный уровень MLS, а не уровень по умолчанию для пользователя.
get_default_context_with_role	То же, что и get_default_context (3), но возвращается только контекст с указанной ролью.
get_default_context_with_rolelevel	То же, что и get_default_context (3), но возвращается только контекст с указанной ролью и уровнем.

Название функции	Описание назначения
get_default_type	Получает тип по умолчанию (домен) для «роли» и устанавливает для него «тип». Вызывающий метод должен освобождаться через free(3). Возвращает 0 при успехе или -1 в противном случае.
getexeccon	Получает контекст exec и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст exec, т. е. используется значение по умолчанию.
getexeccon_raw	Получает контекст exec и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст exec, т. е. используется значение по умолчанию.
getfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getfilecon_raw	Wrapper для xattr API. Получает контекст файла и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getfscreatecon	Получает контекст fscreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если fs не создавал контекст, т. е. используется по умолчанию.
getfscreatecon_raw	Получает контекст fscreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если fs не создавал контекст, т. е. используется по умолчанию.
getkeycreatecon	Получает контекст keycreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст создания ключа, т. е. используется значение по умолчанию.
getkeycreatecon_raw	Получает контекст keycreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст создания ключа, т. е. используется значение по умолчанию.
get_ordered_context_list	Получает упорядоченный список разрешенных контекстов безопасности для сеанса пользователя и устанавливает *conary для ссылки на набор контекстов с завершающим действием NULL. Каждая запись в списке будет разрешена политикой, но запись зависит от пользовательских настроек. Возвращает количество записей в *conary. Если ’fromcon’ равно NULL, по умолчанию используется текущий контекст. Вызывающий метод должен освобождаться через freeconary(3).
get_ordered_context_list_with_level	То же, что и get_ordered_context_list (3), но используется предоставленный уровень MLS, а не уровень по умолчанию для пользователя.
getpeercon	Wrapper для сокета API. Получает контекст

Название функции	Описание назначения
	одноранговой сети и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освобождаться через freecon (3).
getpeercon_raw	Wrapper для сокета API. Получает контекст одноранговой сети и устанавливает для параметра *con ссылку на него. Вызывающий метод должен освобождаться через freecon (3).
getpidcon	Получает контекст процесса, идентифицированный pid, и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getpidcon_raw	Получает контекст процесса, идентифицированный pid, и устанавливает *con для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
getprevcon	Получает предыдущий контекст (до последнего exec) и устанавливает *con для ссылки на него. Звонящий должен освободить через freecon (3).
getprevcon_raw	Получает предыдущий контекст (до последнего exec) и устанавливает *con для ссылки на него. Звонящий должен освободить через freecon (3).
getseuser	Получает имя пользователя и уровень SELinux для использования Linux данного имени и службы. Эти значения затем могут быть переданы в функции get_ordered_context_list* и get_default_context* для получения контекста пользователя. Возвращает 0 при успехе или -1 в противном случае. Вызывающий метод должен освободить возвращенные строки через free(3).
getseuserbyname	Получает имя пользователя и уровень SELinux для использования Linux данного имени пользователя . Эти значения затем могут быть переданы в функции get_ordered_context_list* и get_default_context* для получения контекста пользователя. Возвращает 0 при успехе или -1 в противном случае. Вызывающий метод должен освободить возвращенные строки через free(3).
getsockcreatecon	Получает контекст sockcreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст сокета, т. е. используется по умолчанию. Если не NULL, Вызывающий метод должен освободиться через freecon(3).
getsockcreatecon_raw	Получает контекст sockcreate и устанавливает *con для ссылки на него. Устанавливает *con в NULL, если не задан контекст сокета, т. е. используется по умолчанию. Если не NULL, Вызывающий метод должен освободиться через freecon(3).
_init (если _initselinuxmnt - то вот)	Для этого есть справочная страница, однако она не является доступной пользователю функцией (только для внутреннего использования - хотя доступ к

Название функции	Описание назначения
	fini_selinuxmnt доступен).
is_context_customizable	Возвращает, настраивается ли контекст файла, и его нельзя перемаркировать.
is_selinux_enabled	Возвращает 1, если выполняется в ядре SELinux, или 0, если нет, или -1 в случае ошибки.
is_selinux_mls_enabled	Возвращает 1, если мы работаем на ядре SELinux MLS, или 0 в противном случае.
lgetfilecon	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освободиться через freecon (3).
lgetfilecon_raw	Wrapper для xattr API. Получает контекст файла и устанавливает для него *con. Вызывающий метод должен освободиться через freecon (3).
lsetfilecon	Wrapper для xattr API. Устанавливает контекст файла для символической ссылки.
lsetfilecon_raw	Wrapper для xattr API. Устанавливает контекст файла для символической ссылки.
manual_user_enter_context	Разрешает пользователю вручную вводить контекст в качестве резерва, если список разрешенных контекстов не может быть получен. Вызывающий метод должен освободиться через freecon (3). Возвращает 0 при успехе или -1 в противном случае.
matchmediacon	Совмещает указанный носитель и конфигурацию контекста мультимедиа и устанавливает *con для ссылки на итоговый контекст. Вызывающий метод должен освобождаться через freecon.
matchpathcon	Сопоставляет указанный путь и режим с настройкой контекста файла и устанавливает *con для ссылки на результирующий контекст. «mode» может быть 0, чтобы отключить соответствие режима. Вызывающий метод должен освобождаться через freecon. Если matchpathcon_init (3) еще не был вызван, эта функция вызовет его при первом вызове с NULL-путем.
matchpathcon_checkmatches	Проверяет, не соответствуют ли какие-либо спецификации и сообщениям о них. «Str» используется как префикс для любых предупреждающих сообщений.
matchpathcon_filespec_add	Поддерживает связь между индексом и индексом спецификации и проверяет, связана ли конфликтующая спецификация с тем же inode (например, из-за нескольких жестких ссылок). Если это так, используйте последнюю из двух спецификаций на основе их порядка в конфигурации контекста файла. Возвращает использованный индекс спецификации.

Название функции	Описание назначения
matchpathcon_filespec_destroy	Уничтожает все ассоциации inode, которые были добавлены, например, для перезагрузки новой файловой системы.
matchpathcon_filespec_eval	Отображает статистику использования хэш-таблицы для ассоциаций.
matchpathcon_fini	Освобождает память, выделенную matchpathcon_init.
matchpathcon_index	То же, что и matchpathcon(3), но возвращает индекс спецификации для последующего использования в вызове matchpathcon_filespec_add(3).
matchpathcon_init	Загружает конфигурацию контекста файла, указанную в «path» в памяти, для использования последующими вызовами matchpathcon. Если «path» равен NULL, загружает конфигурацию контекста активного файла, то есть путь, возвращаемый selinux_file_context_path (3). Если флаг MATCHPATHCON_BASEONLY не установлен, эта функция также проверяет файл «path».homedirs и файл «path».local и загружает дополнительные спецификации из них, если они есть.
matchpathcon_init_prefix	То же, что и matchpathcon init (3), но загружает только записи с регулярным выражением, которые имеют ветки которые являются префиксами «prefix».
mode_to_security_class	Переводит mode_t в имя строки класса безопасности (например, S_ISREG = "file").
print_access_vector	Отображает вектор доступа в строчном представлении.
query_user_context	Учитывая список разрешенных контекстов безопасности для пользователя, запрашивает выбор пользователя и устанавливает для него *newcon. Вызывающий метод должен освободиться через freecon (3). Возвращает 0 в случае sucess или -1 в обратном случае.
realpath_not_final	Резрешает все символические ссылки и относительные части имени пути, но НЕ последний компонент (тот же самый realpath (3), если конечный компонент не является символической ссылкой. Итоговым путем должен быть путь размером PATH_MAX + 1.
rpm_execcon	Выполняет вспомогательную операцию для rpm в соответствующем контексте безопасности.
security_av_perm_to_string	Преобразовывает разрешения на доступ к именам строк.
security_av_string	Возвращает вектор доступа в строчном представлении. Пользователь должен освободить возвращаемую строку через free(3).
security_canonicalize_context	Канонизирует контекст безопасности. Возвращает указатель на каноническую (основную) форму контекста безопасности в canoncon, который

Название функции	Описание назначения
	использует ядро, а не то, что предоставляется приложением приложения userpace в con.
security_canonicalize_context_raw	Канонизирует (сохраняет) контекст безопасности. Возвращает указатель на каноническую (основную) форму контекста безопасности в canoncon, который использует ядро, а не то, что предоставляется пользовательским приложением в con.
security_check_context	Проверяет валидность контекста безопасности.
security_check_context_raw	Проверяет валидность контекста безопасности.
security_class_to_string	Преобразует значения класса безопасности в имена строк.
security_commit_booleans	Применяет значения логических параметров
security_compute_av	Вычисляет решение о доступе. Запрашивает, разрешает ли политика доступов исходный контекстный scon для доступа к целевому контексту tcon через класс tclass с запрошенным вектором доступа. Решение возвращается в avd.
security_compute_av_flags	Вычисляет решение доступа и возвращает флаги. Запрашивает, разрешает ли политика доступы исходного контекста scon к целевому контексту tcon через класс tclass с вектором запрошенного доступа. Решение возвращается в avd. который имеет дополнительный флаг. В настоящее время единственным определяемым флагом является SELINUX_AVD_FLAGS_PERMISSIVE, который указывает, что решение было вычислено в разрешающем домене (то есть, язык разрешающей политики был использован в политике или семанаге(8), который использовался для установки домена в разрешающем режиме). Обратите внимание, что это не означает, что SELinux работает в разрешающем режиме, а только в домене scon.
security_compute_av_flags_raw	Вычисляет решение доступа и возвращает флаги. Запрашивает, разрешает ли политика доступы исходного контекста scon к целевому контексту tcon через класс tclass с вектором запрошенного доступа. Решение возвращается в avd. который имеет дополнительный флаг. В настоящее время единственным определяемым флагом является SELINUX_AVD_FLAGS_PERMISSIVE, который указывает, что решение было вычислено в разрешающем домене (то есть, язык разрешающей политики был использован в политике или семанаге(8), который использовался для установки домена в разрешающем режиме). Обратите внимание, что это не означает, что SELinux работает в разрешающем режиме, только в домене scon.

Название функции	Описание назначения
security_compute_av_raw	Вычисляет решение о доступе. Запрашивает, разрешает ли политика доступов исходный контекстный scon для доступа к целевому контексту tcon через класс tclass с запрошенным вектором доступа. Решение возвращается в avd.
security_compute_create	Вычисляет решение для маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon(3).
security_compute_create_name	Функция идентична security_compute_create(3), но также принимает имя нового объекта при создании в качестве аргумента. Когда правило [#5.7.6.type_transition |outline type_transition] </tt> для данного класса, а пара scon/tcon имеет расширение имени объекта, newcon будет возвращаться в соответствии с политикой. Обратите внимание, что этот интерфейс поддерживается только на ядрах 2.6.40 или новее. Для старых ядер имя объекта игнорируется.
security_compute_create_name_raw	Функция идентична security_compute_create(3), но также принимает имя нового объекта при создании в качестве аргумента. Когда правило [#5.7.6.type_transition |outline type_transition] </tt> для данного класса, а пара scon/tcon имеет расширение имени объекта, newcon будет возвращаться в соответствии с политикой. Обратите внимание, что этот интерфейс поддерживается только на ядрах 2.6.40 или новее. Для старых ядер имя объекта игнорируется.
security_compute_create_raw	Вычисляет решение для маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon(3).
security_compute_member	Вычисляет решение для члена polyinstantiation и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_member_raw	Вычисляет решение для члена polyinstantiation и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_relabel	Вычисляет решение для повторной маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).
security_compute_relabel_raw	Вычисляет решение для повторной маркировки и устанавливает *newcon для ссылки на него. Вызывающий метод должен освобождаться через freecon (3).

Название функции	Описание назначения
security_compute_user	Вычисляет набор доступных контекстов пользователя и устанавливает *con для ссылки на массив контекстов с завершающим действием NULL. Вызывающий метод должен освобождаться через freeconary(3).
security_compute_user_raw	Вычисляет набор доступных контекстов пользователя и устанавливает *con для ссылки на массив контекстов с завершающим действием NULL. Вызывающий метод должен освобождаться через freeconary(3).
security_deny_unknown	Получает поведение для неопределенных классов / разрешений.
security_disable	Отключает SELinux во время выполнения (необходимо выполнить до начальной загрузки политики).
security_get_boolean_active	Получает активное значение для логического.
security_get_boolean_names	Получает логические имена
security_get_boolean_pending	Получает ожидающее значение для логического.
security_getenforce	Получает значение флага принудительного исполнения.
security_get_initial_context	Получает контекст исходного идентификатора безопасности ядра по имени. Вызывающий метод должен освободить через freecon (3).
security_get_initial_context_raw	Получает контекст исходного идентификатора безопасности ядра по имени. Вызывающий метод должен освободить через freecon (3).
security_load_booleans	Загрузите логические настройки политики. Путь может быть NULL, и в этом случае логические значения загружаются из активного файла конфигурации логической политики.
security_load_policy	Загружает конфигурацию политики.
security_policyvers	Получает номер версии политики.
security_set_boolean	Устанавливает ожидающее значение для булева.
security_set_boolean_list	Сохраняет список логических элементов в одной транзакции.
security_setenforce	Устанавливает значение флага принудительного исполнения.
selabel_close	Уничтожает указанный дескриптор, закрывает файлы, освобождает выделенную память и т. д. Хендл после закрытия больше не может использоваться.
selabel_cmp	Сравнивает два дескриптора
selabel_digest	Получает хэш дескриптора
selabel_lookup	Выполняет операцию поиска меток. Возвращает 0 при успехе, -1 с errno, устанавливающемся при сбое. Аргументы key и type являются входными (данными) для операции поиска; соответствующие значения продиктованы используемым бэкэндом.

Название функции	Описание назначения
	Результат возвращается в точку памяти, на которую указывает con, и он должен быть освобожден с помощью freecon.
selabel_lookup_best_match	Выбирает одну наиболее подходящую условию поиска метку
selabel_lookup_best_match_raw	Выбирает одну наиболее подходящую условию поиска метку
selabel_lookup_raw	Выполняет операцию поиска меток. Возвращает 0 при успехе, -1 с errno, устанавливающемся при сбое. Аргументы key и type являются входными (данными) для операции поиска; соответствующие значения продиктованы используемым бэкэндом. Результат возвращается в точку памяти, на которую указывает con, и который должен быть освобожден с помощью freecon.
selabel_open	Создает дескриптор маркировки. Откройте бэкэнд для использования маркировки. Доступными бэкэнд идентификаторами являются: SELABEL_CTX_FILE - file_contexts. SELABEL_CTX_MEDIA - media contexts. SELABEL_CTX_X - x_contexts. SELABEL_CTX_DB - contexts SE-PostgreSQL. SELABEL_CTX_ANDROID_PROP - property_contexts. Параметры могут предоставляться с помощью параметра opts; доступны следующие варианты: SELABEL_OPT_UNUSED - опция no-op, полезная для неиспользуемых слотов в массиве опций. SELABEL_OPT_VALIDATE - проверять контексты перед их возвратом (логическое значение). SELABEL_OPT_BASEONLY - не используйте локальные настройки для создания бэкэнд-данных (логическое значение). SELABEL_OPT_PATH - указать альтернативный путь для использования при загрузке данных на бэкэнд. SELABEL_OPT_SUBSET - выберите подмножество поискового пространства в качестве оптимизации (файловый сервер). Не все опции могут поддерживаться каждым бэкэнд. Возвращаемое значение - это успешно созданный дескриптор или NULL с errno, установленным при ошибке. "
selabel_partial_match	Выбирает одну наиболее подходящую условию поиска метку
selabel_stats	Логирует сообщение с информацией о количестве выполненных запросов, количестве неиспользуемых совпадающих записей или другой рабочей статистике. Сообщение является специфичным для

Название функции	Описание назначения
	бэкэнда, некоторые серверы могут не выводить сообщение.
selabel_subs_init	Инициализрует работу с метками
selinux_binary_policy_path	Возвращает путь к двоичному файлу политики из корневого каталога политики.
selinux_booleans_path	Возвращает путь к логическому файлу в корневом каталоге политики.
selinux_booleans_subs_path	Возвращает путь к файлу конфигурации booleans.subs_dist.
selinux_boolean_sub	Считывает файл /etc/selinux/TYPE/booleans.subs_dist, ища запись с именем boolean_name. Если существует запись, selinux_boolean_sub (3) возвращает переведенное имя, иначе возвращается исходное имя. Возвращаемое значение должно быть освобождено. При ошибке будет возвращен NULL.
selinux_check_access	Используется для проверки того, имеет ли исходный контекст разрешение доступа для указанного класса в целевом контексте. Обратите внимание, что разрешение и класс являются ссылочными строками. Параметр aux может ссылаться на дополнительную информацию аудита. Аудит обрабатывается, как описано в avc_audit (3). См. Security_deny_unknown (3) о том, как флаг deny_unknown может влиять на решения политики.
selinux_check_passwd_access	Проверяет разрешение в классе passwd. Возвращает 0, если предоставлено или -1 в противном случае. Заменено selinux_check_access (3)
selinux_check_securetty_context	Проверяет, что tty_context определяется как securetty. Возвращает 0, если безопасно, <0 в противном случае.
selinux_colors_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_contexts_path	Возвращает путь к каталогу контекстов в корневом каталоге политики.
selinux_current_policy_path	Возвращает путь к текущей политике.
selinux_customizable_types_path	Возвращает путь к файлу customizable_types в корневом каталоге политики.
selinux_default_context_path	Возвращает к файлу default_context в корневом каталоге политики.
selinux_default_type_path	Возвращает путь к файлу default_type.
selinux_failsafe_context_path	Возвращает путь к файлу failafe_context в корневом каталоге политики.
selinux_file_context_cmp	Сравнивает два контекста файлов, возвращайте 0, если они эквивалентны.
selinux_file_context_homedir_path	Возвращает путь к файлу file_context.homedir в корневом каталоге политики.

Название функции	Описание назначения
selinux_file_context_local_path	Возвращает путь к файлу file_context.local в корневом каталоге политики.
selinux_file_context_path	Возвращает путь к файлу file_context в корневом каталоге политики.
selinux_file_context_subs_dist_path	Возвращает путь к файлу file_context.subs_dist в корневом каталоге политики.
selinux_file_context_subs_path	Возвращает путь к файлу file_context.subs в корневом каталоге политики.
selinux_file_context_verify	Проверяет контекст файла "path" к политике. Возвращает 0, если это правильно.
selinuxfs_exists	Проверка существования псевдофайловой системы SELinux
selinux_get_callback	Используется для получения указателя на функцию обратного вызова данного типа. Функции обратного вызова устанавливаются с помощью selinux_set_callback (3).
selinux_getenforcemode	Читает файл /etc/selinux/config и определяет, следует ли запускать машину в режиме принудительного (1), разрешающего (0) или отключенного (-1) режима.
selinux_getpolicytype	Читает файл / etc / selinux / config и определяет, что такое политика по умолчанию для машины. Вызов приложения должен освобождаться policytype.
selinux_homedir_context_path	Возвращает путь к файлу в корневом каталоге политики. Обратите внимание, что этот файл будет отображаться только в более старых версиях политики в этом месте. В системах, управляемых с помощью semanage(8), это теперь находится в хранилище политик.
selinux_init_load_policy	Выполняет начальную загрузку политики. Эта функция определяет желаемый режим принудительного исполнения, устанавливает аргумент *enforce (принудительного) использования для Вызывающий метод, устанавливает соответствие статуса принудительного использования ядра SELinux и загружает политику. Он также внутренне обрабатывает начальное монтирование selinuxfs, необходимое для выполнения этих действий. Функция возвращает 0, если все, включая загрузку политики, завершается успешно. В этом случае init, как ожидается, будет повторно выполнять сам, чтобы перейти к надлежащему контексту безопасности. В противном случае функция возвращает -1, и init должен проверить *enforce, чтобы определить, как действовать. Если принудительное выполнение (*inforce> 0), то init должен остановить систему. В противном случае init может нормально работать без повторного запуска.

Название функции	Описание назначения
selinux_lsetfilecon_default	Эта функция устанавливает контекст файла в системные значения по умолчанию. Возвращает 0 при успехе.
selinux_lxc_contexts_path	Возвращает путь к файлу конфигурации lxc_contexts.
selinux_media_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_mkload_policy	Создает слепок политики и загружает его. Эта функция обеспечивает интерфейс более высокого уровня для загрузки политики, чем security_load_policy (3), внутренне определяет правильную версию политики, устанавливает и открывает файл политики, сопоставляет ее в памяти, манипулируя ею (памятью) по мере необходимости для текущих логических настроек и/или локальных определений и затем вызывает security_load_policy (3), чтобы загрузить ее. «preservebools» - это логический флаг, указывающий, следует ли сохранять текущие логические значения политики в новой политике (если 1) или сбросить сохраненные параметры политики (если 0). Первый случай является значением по умолчанию для перезагрузки политики, в то время как последний случай является вариантом для перезагрузки политики, но в первую очередь для начальной загрузки политики.
selinux_netfilter_context_path	Возвращает путь к файлу netfilter_context в корневом каталоге политики.
selinux_openssh_contexts_path	Возвращает путь к файлу openssh_context в корневом каталоге политики.
selinux_path	Возвращает путь к корневому каталогу политики.
selinux_policy_root	Считывает файл /etc/selinux/config и возвращает каталог верхнего уровня.
selinux_raw_context_to_color	Выполняет контекстный перевод между контекстами безопасности и цветами отображения. Возвращает список разделенных пробелами десяти шестнадцатеричных триггеров RGB с префиксом хэш-меток, например. "#ff0000". Вызывающий метод должен освободить полученную строку через free(3). Возвращает -1 при ошибке или 0 в противном случае.
selinux_raw_to_trans_context	Выполняет контекстный перевод между человеко-понятным форматом («translated») и внутренним системным форматом («raw»). Вызывающий метод должен освободить полученный контекст через freecon(3). Возвращает -1 при ошибке или 0 в противном случае. Если передано NULL, задается возвращаемый контекст в NULL и возвращает 0.

Название функции	Описание назначения
selinux_removable_context_path	Возвращает путь к файлу removeable_context в корневом каталоге политики.
selinux_reset_config	Принудительный сброс загруженной конфигурации.
selinux_restorecon	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_default_handle	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_set_exclude_list	Восстанавливает соединение с псевдофайловой системой
selinux_restorecon_set_sehandle	Восстанавливает соединение с псевдофайловой системой
selinux_securetty_types_path	Возвращает путь к файлу securetty_types в корневом каталоге политики.
selinux_sepgsql_context_path	Возвращает путь к файлу sepgsql_context в корневом каталоге политики.
selinux_set_callback	Устанавливает обратный вызов в соответствии с типом: SELINUX_CB_LOG, SELINUX_CB_AUDIT, SELINUX_CB_VALIDATE, SELINUX_CB_SETENFORCE, SELINUX_CB_POLICYLOAD
selinux_set_mapping	Поддержка сопоставления класса пространства пользователей, которая устанавливает сопоставление от пользовательского заказа классов объектов и разрешений на номера, фактически используемые загруженной системной политикой.
selinux_set_policy_root	Устанавливает альтернативный путь корневого каталога политики, в соответствии с которым существуют файлы скомпилированного файла политики и файлы конфигурации контекста.
selinux_snapperd_contexts_path	Устанавливает путь до базовых контекстов
selinux_status_close	Отменяет и закрывает страницу состояния ядра.
selinux_status_deny_unknown	Получает поведение для неопределенных классов/разрешений.
selinux_status_getenforce	Получает значение флага принудительного исполнения.
selinux_status_open	Открывает и отображает страницу состояния ядра SELinux.
selinux_status_policyload	Получает количество загрузок политики.
selinux_status_updated	Сообщает, обновлен ли статус ядра.
selinux_systemd_contexts_path	Возвращает путь к файлу конфигурации systemd_contexts.
selinux_translations_path	Возвращает путь к файлу setrans.conf в корневом каталоге политики.
selinux_trans_to_raw_context	Выполняет контекстный перевод между человеко-читаемым форматом («translated») и внутренним системным форматом («raw»). Вызывающий метод должен освободить полученный контекст через freecon(3). Возвращает -1 при ошибке или 0 в

Название функции	Описание назначения
	противном случае. Если передано NULL, задается возвращаемый контекст в NULL и возвращает 0.
selinux_user_contexts_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_usersconf_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_users_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_virtual_domain_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_virtual_image_context_path	Возвращает путь к файлу в корневом каталоге политики.
selinux_x_context_path	Возвращает путь к файлу x_context в корневом каталоге политики.
setcon	Устанавливает для текущего контекста безопасности значение con. Обратите внимание, что для использования этой функции требуется, чтобы всему приложению было доверено поддерживать любое разделение между старыми и новыми контекстами безопасности, в отличие от переходов на основе exec, выполняемых через setexeccon (3). Когда это возможно, разложите приложение и используйте setexeccon (3) + execve (3). Обратите внимание, что приложение может потерять доступ к своим открытым дескрипторам в результате setcon (3), если только политика не позволяет использовать дескрипторы, открытые старым контекстом.
setcon_raw	Устанавливает для текущего контекста безопасности значение con. Обратите внимание, что для использования этой функции требуется, чтобы всему приложению было доверено поддерживать любое разделение между старыми и новыми контекстами безопасности, в отличие от переходов на основе exec, выполняемых через setexeccon (3). Когда это возможно, разложите приложение и используйте setexeccon (3) + execve (3). Обратите внимание, что приложение может потерять доступ к своим открытым дескрипторам в результате setcon (3), если только политика не позволяет использовать дескрипторы, открытые старым контекстом.
setexeccon	Устанавливает контекст безопасности exec для следующего execve (3). Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setexeccon_raw	Устанавливает контекст безопасности exec для следующего execve (3). Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setexecfilecon	Устанавливает соответствующий контекст

Название функции	Описание назначения
	безопасности, основанный на имени файла вспомогательной программы, возвращаясь к новому контексту с указанным типом.
setfilecon	Обертка для xattr API - устанавливает контекст файла.
setfilecon_raw	Обертка для xattr API - устанавливает контекст файла.
setfscreatecon	Устанавливает контекст безопасности fscreate для последующего создания файлов. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setfscreatecon_raw	Устанавливает контекст безопасности fscreate для последующего создания файлов. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setkeycreatecon	Устанавливает контекст безопасности keycreate для последующих создания ключей. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
setkeycreatecon_raw	Устанавливает контекст безопасности keycreate для последующих создания ключей. Вызов с NULL, если вы хотите сбросить к значению по умолчанию.
set_matchpathcon_canoncon	То же, что и set_matchpathcon_invalidcon (3), но также позволяет канонизацию контекста, изменяя * context на ссылку на каноническую форму. Если он не установлен, а invalidcon также не установлен, то по умолчанию используется вызов security_canonicalize_context (3).
set_matchpathcon_flags	Устанавливает флаги, управляющие работой matchpathcon_init (3) или matchpathcon (3): MATCHPATHCON_BASEONLY - обрабатывает только файл base_contexts. MATCHPATHCON_NOTRANS - не выполняет никакого контекстного перевода. MATCHPATHCON_VALIDATE - проверяет/канонизирует контексты вовремя init.
set_matchpathcon_invalidcon	Устанавливает функцию, используемую matchpathcon_init (3) при проверке действительности контекста в конфигурации file_contexts. Если это не задано, то по умолчанию используется тест на основе security_check_context (3). Эта функция также отвечает за сообщение о любой такой ошибке и может включать в себя такие сообщения, как «path» и «lineno».
set_matchpathcon_printf	Установливает функцию, используемую matchpathcon_init (3) при отображении ошибок в конфигурации file_contexts. Если не задано, то по умолчанию используется fprintf (stderr, fmt, ...).
set_selinuxmnt	Явно устанавливает путь к точке монтирования selinuxfs. Обычно это определяется автоматически во время инициализации libselinux, но это не всегда возможно, например, для/sbin/init, который выполняет начальную установку selinuxfs.

Название функции	Описание назначения
setsockcreatecon	Устанавливает контекст безопасности sockcreate для последующих созданий сокетов. Вызовите с NULL, если вы хотите сбросить к значению по умолчанию.
setsockcreatecon_raw	Устанавливает контекст безопасности sockcreate для последующих созданий сокетов. Вызовите с NULL, если вы хотите сбросить к значению по умолчанию.
sidget	Из 2.0.86 - не поддерживается
sidput	Из 2.0.86 - не поддерживается
string_to_av_perm	Преобразование имен строк для доступа к векторным разрешениям.
string_to_security_class	Преобразование имен строк в значения классов безопасности.

ПРИЛОЖЕНИЕ Б. ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

(обязательное)

АИС	–	автоматизированная информационная система
АРМ	–	автоматизированное рабочее место
АСЗИ	–	автоматизированная система в защищенном исполнении
НСД	–	несанкционированный доступ
ПК	–	программный комплекс
ПС	–	программное средство
ПСЗИ	–	программное средство защиты информации
СХД	–	система хранения данных

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	Номер документа	Входящий номер сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					